**Programming Assignment 1**

**Name:** Sai Naga Viswa Chaitanya Basava

**NetID:** SXB220302

**Email:** [sxb220302@utdallas.edu](mailto:sxb220302@utdallas.edu)

Experiments were executed on a 2019 MacBook pro with **12** 2.6 GHz 6-Core Intel Core i7 processors.

Each algorithm was run for a thread count range of 2 to 12, with 50 runs for each experiment. The shared counter is incremented for a total of 1 million times in total, with each thread requesting for the critical section for an equal number of times. The outcome of the experiments is as following:

A graph of different colored lines

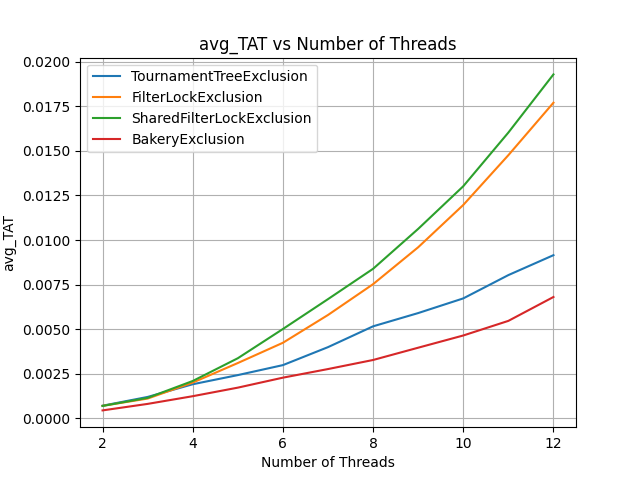
Description automatically generated

From the throughput graph we can observe the following trend w.r.t number of threads

The 2 versions of Filter Locks had almost the same performance, with the level based version with padding to avoid false sharing (class version) showing slightly better performance than the shared memory version as the number of threads increase, this can be accounted to the RMR differences, similar to which was discussed in the class.

The tournament based algorithm had better performance compared to both these versions as the number of threads increased, which shows the impact of how logN level comparisons can perform better than the version where it had to go through N-1 levels to acquire the lock.

The Bakery algorithm had the best performance in terms of throughput but had also showed a faster rate of decrease in throughput as compared to the other algorithms, which had a much steady decrease in throughput as N increased.



Bakery algorithm depicts the lowest TAT and also increases at a steady pace with the number of threads. Like in the throughput graph, both the versions of the File lock algorithms perform close to each other and have the fastest rate of increase of TAT as the thread count increases. The version discussed in the class with padding has slightly lower TAT compared to the textbook version, which further depicts the impact of RMR in practice.

As observed from the throughput graph, the power of logN levels compared to N-1 levels is highlighted in this graph as well. The tournament tree algorithm showed steady growth in TAT and is much lower for 12 threads.

**Summary**

In summary, Bakery algorithm was the best performing of all the implemented algorithms both in terms of throughput and average Turn Around Time.